Optional Projects **CMPT 238 Data Structures**

Due: before the last two weeks of the semester **Fall 2024**

Rules and Grading:

1. You must work individually.
2. For each project, you must use two searching algorithms, at least two sorting algorithms, and at least two different data structures (arrays are not allowed). The more algorithms and data structures you use the more credits you will get
3. Submit the followings to Moodle (lms.manhattan.edu):
   - Readme.txt (Readme.doc) should contain the project number and at least two test runs with their input and output. You should also include necessary explanations to any data structures or algorithms/functions used in the code.
   - Zip package of source code, executable files, and the Readme file submit it to Moodle.
4. The evaluation will be in person before the last two weeks of the semester.
5. I will ask you to run the code in front of me and I will ask questions about the code, so be prepared.
6. Any copied code from the internet will be considered cheating and there will be no tolerance for this, please don't do it.
7. **The assigned credits for each project are listed to the next of the project name, BUT, in order to get these credits, you must solve a complete and correct project. These credits will be added to your midterm exam grade or to one of your assignments grade, you can solve more than one project to get more credits.**
8. You can suggest a project to work on rather than the projects listed below, however, you need my approval before you work on the suggested project.

Projects:

1. Building a Phonebook (10 points)

a) Use a proper data structure to store contact records.
b) Implement functions to add, remove, and search for contacts by name or number.

c) Use sorting algorithms to arrange contacts alphabetically.
d) Implement search algorithms like binary search or linear search for efficient contact lookup.
e) Provide a user interface for adding, editing, deleting, and searching contacts in the phonebook.

2. Banking management system (15 points)

a. Identify core banking operations like account management, transaction handling, reporting, etc.
b. Choose appropriate data structures, such as arrays/lists/vectors for accounts.
c. Implement account management: create, update, and delete accounts.
d. Handle transactions like deposits, withdrawals, and money transfers using queues/stacks.
e. Use sorting algorithms for report generation for account balances.
f. Implement search algorithms for account lookup.
g. Ensure data integrity with error handling and input validation.

3. Building a crossword puzzle game (15 points)

This program will train students to build a crossword puzzle game that can generate and solve various puzzles. It will take a list of words, generate a crossword grid, and place the words in an interlocking pattern. It will also help users find and fill in the missing words depending on clues and the interlocking pattern of the grid.

a. Use a 2D array or matrix to represent the crossword puzzle grid.
b. Integrate randomization for puzzle generation and variation.
c. Implement data structures like stacks or queues for undo/redo functionality.
d. Provide a user interface for displaying the puzzle, entering words, and interacting with game features.
e. Handle input validation, error checking, and scoring mechanisms for gameplay.

4. Library management system (15 points)

This project can efficiently manage library operations like cataloging books, tracking book borrowing and returns, member record maintenance, generating

a. Use a data structure to store book records (title, author, availability).
b. Implement functions to add, remove, and search for books by title, author, or other criteria.
c. Use sorting algorithms to arrange books alphabetically or by other attributes.
d. Implement search algorithms for efficient book lookup.
e. Implement data structures like queues or priority queues for handling book reservations and waitlists.
f. Handle member management, book lending/return.
g. Provide a user interface for librarians and members to interact with the system.

5. Snakes Game (10 Points)

- The snake game is one of the most popular games on Nokia phones, released in 1998

- Build a snake game using arrays and linked lists

6. Simulate scheduling algorithms (like Round Robin, Shortest Job First) for process scheduling in operating systems. (10 points)

7. Use data structures and algorithms to plan paths for robots in constrained environments, avoiding obstacles. (15 points)

8. Implement a system to manage orders, tables, and inventory in a restaurant. (15 points)

9. Healthcare Information System: Develop a system to manage patient records, appointments, and medical histories. (15 Points)

10. Flight Reservation System: Create a system using queues or priority queues to manage flight bookings, seats, and schedules. (15 Points)

11. Document Similarity Checker: Implement algorithms and data structures to determine the similarity between documents or texts. (20 points)

12. Parking Management System: Develop a system using queues or arrays to manage parking slots and vehicle entry/exit in parking lots. (10 Points)